

# Delay-Optimal Buffer-Aware Probabilistic Scheduling with Adaptive Transmission

Xiang Chen, *Student Member, IEEE*, Wei Chen, *Senior Member, IEEE*

Tsinghua National Laboratory for Information Science and Technology (TNList)

Department of Electronic Engineering, Tsinghua University, Beijing, 100084, CHINA

Email: chen-xiang12@mails.tsinghua.edu.cn, wchen@tsinghua.edu.cn

**Abstract**—Cross-layer scheduling is a promising way to improve Quality of Service (QoS) given a power constraint. In this paper, we investigate the system with random data arrival and adaptive transmission. Probabilistic scheduling strategies aware of the buffer state are applied to generalize conventional deterministic scheduling. Based on this, the average delay and power consumption are analysed by Markov reward process. The optimal delay-power tradeoff curve is the Pareto frontier of the feasible delay-power region. It is proved that the optimal delay-power tradeoff is piecewise-linear, whose vertices are obtained by deterministic strategies. Moreover, the corresponding strategies of the optimal tradeoff curve are threshold-based, hence can be obtained by a proposed effective algorithm. On the other hand, we formulate a linear programming to minimize the average delay given a fixed power constraint. By varying the power constraint, the optimal delay-power tradeoff curve can also be obtained. It is demonstrated that the algorithm result and the optimization result match each other, and are further validated by Monte-Carlo simulation.

## I. INTRODUCTION

Real-time services such as instant messenger (IM), social network service (SNS) and streaming media are experiencing a period of rapid growth, therefore are specially concerned by next generation networks. Services of this sort all have high requirements for manipulated delay. However, due to the randomness of the workload and the architecture of traditional packet-switched communication network, delay is not easy to analyse or control. It is widely acknowledged that there is a fundamental tradeoff between the average queueing delay and the power consumption. Therefore to achieve better QoS, cross-layer optimization with traffic and energy management is well studied and accepted.

One of the major methods of achieving the optimal delay is dynamic programming. To our best knowledge, [1] is one of the earliest work to study cross-layer scheduling model considering a time-varying channel and average delay constraints. Berry and Gallager investigated adapting the user's transmission rate and power over a fading channel in [2]. The optimal power-delay tradeoff was achieved using Markov decision process formulation and dynamic programming. Based on this, the existence of stationary optimal scheduling as well as the lower and upper bounds of the optimal policy was discovered in [3]. Detailed power/rate adaptation was further studied in [4], where analytic analysis for asymptotically large buffer size was conducted. In [5], Ata modelled the problem as a continuous-time Markov chain under the assumption of fixed

channel, Poisson arrival and exponentially distributed packet size. Instead of considering average delay, hard deadline as well as static channel assumption was considered in [6], where lazy scheduling was proposed to minimize the total energy consumption. A cumulative curves methodology was investigated in [7] so that energy-efficient transmission with deadline constraint and continuous-time optimization can be achieved. In [8], closed-form optimal policy for two timeslots and numerical optimal policy for more timeslots were obtained, which were found to be a linear combination of a delay-associated term and a channel-aware term. K-block delay constraint with causal feedback of the channel state was imposed in [9].

In our previous work [10], we investigated the situation where data packets arrive randomly and the channel is time-varying. We proposed a cross-layer method to achieve closed-form delay-power tradeoff and optimal scheduling strategy by a fixed-modulation scheme. Furthermore, in order to apply our formulation to various problems, the long-coherent-time and short-coherent-time cases were studied respectively in [11] and [12]. All the above work is based on simple communication protocols with fixed modulation and coding scheme so that the computation and energy cost can be saved. However, adaptive transmission rate is widely supported for many application scenarios. Therefore we advance a new approach to analysing the system with adaptive transmission in this paper.

The remainder of this paper is organized as follows. The system model is established in Section II. We formulate the problem as a Markov reward process in Section III, based on which the average power consumption and the average delay are analysed and expressed by steady-state probability distribution. In Section IV we investigate some properties of the scheduling strategies. It is discovered that the feasible delay-power region is a polyhedron, whose vertices are obtained by deterministic scheduling. The optimal delay-power tradeoff curve, as the Pareto frontier of the polyhedron, is piecewise linear. Moreover, the optimal scheduling is threshold-based, thus an algorithm is proposed to obtain the optimal tradeoff curve. In Section V, we formulate a linear programming to minimize the average delay given an average power constraint. By varying the power constraint, the optimal tradeoff curve can also be acquired. It is demonstrated that the algorithm, the optimization and the simulation results completely match each other in Section VI. Section VII concludes the paper.

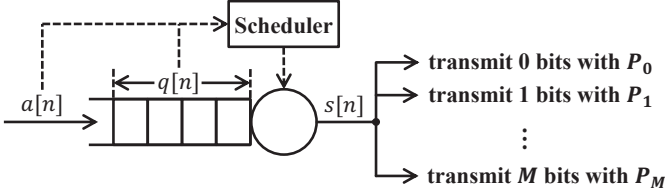


Fig. 1. System Model

## II. SYSTEM MODEL

Consider the system model shown in Fig.1. We assume that at the beginning of each timeslot, data arrive as a Bernoulli Process with parameter  $\alpha$ . Each incoming data packet contains  $A$  bits. Denote  $a[n] = 1$  or  $0$  as whether or not there are data arriving in timeslot  $n$ , hence

$$\begin{cases} \Pr\{a[n] = 1\} = \alpha, \\ \Pr\{a[n] = 0\} = 1 - \alpha. \end{cases} \quad (1)$$

Denote  $s[n]$  as the number of data bits transmitted in timeslot  $n$ . Because of the constraints of the transmitter, in each timeslot at most  $M$  bits can be transmitted. In order to guarantee the stability of the system, we set  $M \geq A$ . Assume transmitting  $m$  bits will cost power  $P_m$ . Transmitting 0 bit will cost no power, hence  $P_0 = 0$ . From perspective of the physical layer, to transmit more bits without increasing the error rate, we should use a larger constellation diagram, therefore more power has to be consumed for every bit in average, i.e.  $\frac{P_{m_1}}{m_1} < \frac{P_{m_2}}{m_2}$  if  $m_1 < m_2$ . More details can be found in [6], where information-theoretic explanations and a few examples are provided to support this point.

To temporarily store the backlog data, a buffer is introduced. Assume the buffer can restore at most  $Q$  bits. Denote  $q[n]$  as the number of bits in the buffer at the beginning of timeslot  $n$ . Therefore in timeslot  $n$ , we can decide how many bits to be transmitted based on the buffer state  $q[n]$  and the data arrival  $a[n]$ . To extend from traditional deterministic scheduling methods so that a better delay-power tradeoff curve can be achieved, we investigate probabilistic scheduling policies. In other words, given  $q[n]$  and  $a[n]$ , the value of  $s[n]$  is determined by a probability distribution. In each timeslot, the transmission is scheduled after the data arrival, thus when deciding how many bits should be transmitted, there are  $(q[n] + Aa[n])$  bits data in total. Denote  $t[n] = q[n] + Aa[n]$  and  $K = Q + A$ . For transmission, there is no need to distinguish the newly arrived data from the backlog data, therefore we denote  $f_{k,m}$  as the probability to transmit  $m$  bits when  $t[n] = k$ , i.e.

$$f_{k,m} = \Pr\{s[n] = m | t[n] = k\}. \quad (2)$$

Immediately we have

$$\sum_{m=0}^M f_{k,m} = 1 \quad \forall k = 0, \dots, K. \quad (3)$$

Denote  $\mathbf{F}$  as a  $(K+1) \times (M+1)$  matrix whose element in the  $(k+1)$ th row and the  $(m+1)$ th column is  $f_{k,m}$ . We guarantee that the transmission strategy will avoid overflow or underflow, which means  $f_{k,m} = 0$  if  $k-m < 0$  or  $k-m > Q$ . Therefore the following equation should hold

$$q[n+1] = q[n] + Aa[n] - s[n], \quad (4)$$

$$t[n+1] = t[n] - s[n] + Aa[n+1]. \quad (5)$$

## III. MARKOV REWARD PROCESS

As described in the above section, the scheduler will decide how many bits to be transmitted based on the information of the queue length and the data arrival state. Since  $a[n]$  is *i.i.d.*,  $s[n]$  is determined by  $t[n]$ , based on (5) we have

$$\Pr\{t[n+1] | t[n], \dots, t[1]\} = \Pr\{t[n+1] | t[n]\}. \quad (6)$$

Hence the stochastic process  $\{t[n]\}$  is Markovian. Therefore we can formulate the problem as a Markov Reward Process (MRP), where the average delay and power are the rewards of the Markov process. Denote  $\pi_k$  as the steady-state probability for  $t[n] = k$ . Denote  $\lambda_{i,j}$  as the transition probability from state  $i$  to state  $j$ . The transition diagram is shown in Fig.2, where  $\lambda_{i,i}$  for  $i = 0, \dots, K$  are omitted to keep the diagram legible. The transition from state  $i$  to state  $j$  is made up of at most two cases. One is transmitting  $(i-j)$  bits and then there are no data arriving. The other is transmitting  $(i-j+A)$  bits and then there are data arriving. Therefore we have

$$\lambda_{i,j} = \begin{cases} (1-\alpha)f_{i,i-j} & M-A < i-j \leq M \\ (1-\alpha)f_{i,i-j} & 0 \leq i-j \leq M-A, j < A \\ (1-\alpha)f_{i,i-j} & 0 \leq i-j \leq M-A, A \leq j \leq K-A \\ +\alpha f_{i,i-j+A} & \\ \alpha f_{i,i-j+A} & 0 \leq i-j \leq M-A, j > K-A \\ \alpha f_{i,i-j+A} & -A \leq i-j < 0, j \geq A \\ 0 & \text{else} \end{cases} \quad (7)$$

Denote  $\boldsymbol{\pi} = [\pi_0, \dots, \pi_K]^T$ . Denote  $\mathbf{A}_F$  as a  $(K+1) \times (K+1)$  matrix whose element in the  $(i+1)$ th column and the  $(j+1)$ th row is  $\lambda_{i,j}$ . It is determined by  $\mathbf{F}$ . Denote  $\mathbf{I}$  as the identity matrix,  $\mathbf{1} = [1, \dots, 1]^T$ , and  $\mathbf{0} = [0, \dots, 0]^T$ . We won't specify their size if there is no ambiguity. Denote  $\mathbf{G}_F = \mathbf{A}_F - \mathbf{I}$ . Therefore we have  $\mathbf{G}_F \boldsymbol{\pi} = \mathbf{0}$  and  $\mathbf{1}^T \boldsymbol{\pi} = 1$ . Since the sum of all rows in  $\mathbf{G}_F$  is  $\mathbf{0}^T$ , we set  $\mathbf{H}_F = \begin{bmatrix} \mathbf{1}^T \\ \mathbf{G}_F(0:(K-1), :) \end{bmatrix}$  and  $\mathbf{c} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ . Hence

$$\boldsymbol{\pi} = \mathbf{H}_F^{-1} \mathbf{c}. \quad (8)$$

In other words,  $\mathbf{H}_F$  can be used to represent a certain transmission strategy. It will determine the value of  $\boldsymbol{\pi}$ .

We can express the average delay and the average power consumption using the steady-state probability distribution. For state  $k$ , transmitting  $m$  bits will cost  $P_m$  with probability

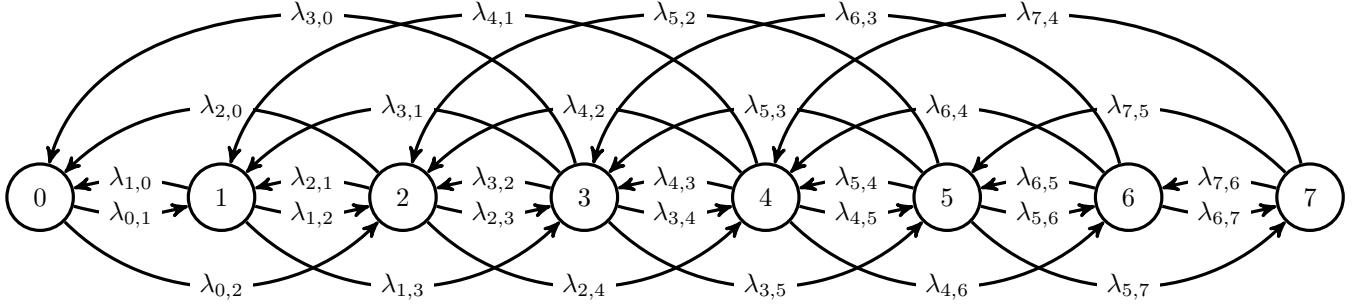


Fig. 2. Markov Chain of  $t[n]$  ( $K = 7$ ,  $A = 2$ ,  $M = 3$ )

$f_{k,m}$ . Denote  $\mathbf{p}_F = [\sum_{m=0}^M P_m f_{0,m}, \dots, \sum_{m=0}^M P_m f_{K,m}]^T$  as a function of  $\mathbf{F}$ , thus the average power consumption

$$P = \sum_{k=0}^K \pi_k \sum_{m=0}^M P_m f_{k,m} = \mathbf{p}_F^T \boldsymbol{\pi}. \quad (9)$$

Similarly, we can obtain that the average queue length is

$$\text{Ex}\{q[n]\} = \text{Ex}\{t[n] - a[n]\} = \sum_{k=0}^K k \pi_k - \alpha A. \quad (10)$$

According to Little's Law, the average delay is the quotient of the average queue length divided by the average arrival rate. Denote  $\mathbf{d} = [0, 1, \dots, K]^T$ , we have

$$D = \frac{1}{\alpha A} \left( \sum_{k=0}^K k \pi_k - \alpha A \right) = \frac{1}{\alpha A} \mathbf{d}^T \boldsymbol{\pi} - 1. \quad (11)$$

#### IV. PROPERTIES OF SCHEDULING STRATEGIES

From above it can be seen that  $\mathbf{p}_F$  and  $\mathbf{H}_F$  are determined by  $\mathbf{F}$ . Moreover,  $\boldsymbol{\pi}$  is determined by  $\mathbf{H}_F$ . Therefore a certain scheduling strategy  $\mathbf{F}$  will determine a delay-power pair  $(P, D)$ . Denote  $(P_F, D_F)$  as the delay-power pair generated by strategy  $\mathbf{F}$ . The mapping from  $\mathbf{F}$  to  $(P_F, D_F)$  has the following property.

**Theorem 1.**  $\mathbf{F}$  and  $\mathbf{F}'$  are two scheduling strategies that are different only when  $t[n] = k$ , i.e. the two matrices are different only in the  $(k+1)$ th row. Denote  $\mathbf{F}'' = (1-\epsilon)\mathbf{F} + \epsilon\mathbf{F}'$  where  $0 \leq \epsilon \leq 1$ . Then

1) There exists a certain  $0 \leq \epsilon' \leq 1$  so that  $P_{F''} = (1-\epsilon')P_F + \epsilon'P_{F'}$  and  $D_{F''} = (1-\epsilon')D_F + \epsilon'D_{F'}$ . Moreover,  $\epsilon'$  is a continuous nondecreasing function of  $\epsilon$ .

2) When  $\epsilon$  changes from 0 to 1, point  $(P_{F''}, D_{F''})$  moves on the line segment from  $(P_F, D_F)$  to  $(P_{F'}, D_{F'})$ .

*Proof:* We will prove the two conclusions one by one.

1) From the definition of  $\mathbf{H}_F$  and  $\mathbf{p}_F$ , we can see that if  $\mathbf{F}'' = (1-\epsilon)\mathbf{F} + \epsilon\mathbf{F}'$ , then  $\mathbf{H}_{F''} = (1-\epsilon)\mathbf{H}_F + \epsilon\mathbf{H}_{F'}$  and  $\mathbf{p}_{F''} = (1-\epsilon)\mathbf{p}_F + \epsilon\mathbf{p}_{F'}$ . Denote  $\Delta\mathbf{H} = \mathbf{H}_{F'} - \mathbf{H}_F$  and  $\Delta\mathbf{p} = \mathbf{p}_{F'} - \mathbf{p}_F$ . Since  $\mathbf{F}$  and  $\mathbf{F}'$  are different only in the  $(k+1)$ th row, it can be derived that  $\Delta\mathbf{H}$  has nonzero element only in the  $(k+1)$ th column, and the  $(k+1)$ th element of  $\Delta\mathbf{p}$  is its only nonzero element. Therefore  $\Delta\mathbf{H}$  can be denoted as

$[\mathbf{0}, \dots, \boldsymbol{\delta}_k, \dots, \mathbf{0}]$ , where  $\boldsymbol{\delta}_k$  is its  $(k+1)$ th column.  $\Delta\mathbf{p}$  can be denoted as  $[0, \dots, \zeta_k, \dots, 0]^T$ , where  $\zeta_k$  is its  $(k+1)$ th

element. Also, we denote  $\mathbf{H}_F^{-1} = \begin{bmatrix} \mathbf{h}_0^T \\ \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_K^T \end{bmatrix}$ . Hence

$$(\mathbf{H}_F^{-1} \Delta\mathbf{H}) \mathbf{H}_F^{-1} = \begin{bmatrix} (\mathbf{h}_0^T \boldsymbol{\delta}_k) \mathbf{h}_k^T \\ (\mathbf{h}_1^T \boldsymbol{\delta}_k) \mathbf{h}_k^T \\ \vdots \\ (\mathbf{h}_K^T \boldsymbol{\delta}_k) \mathbf{h}_k^T \end{bmatrix}. \quad (12)$$

By mathematical induction, we can have that for  $i \geq 1$ ,

$$(\mathbf{H}_F^{-1} \Delta\mathbf{H})^i \mathbf{H}_F^{-1} = \begin{bmatrix} (\mathbf{h}_0^T \boldsymbol{\delta}_k) (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} \mathbf{h}_k^T \\ (\mathbf{h}_1^T \boldsymbol{\delta}_k) (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} \mathbf{h}_k^T \\ \vdots \\ (\mathbf{h}_K^T \boldsymbol{\delta}_k) (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} \mathbf{h}_k^T \end{bmatrix} \quad (13)$$

$$= (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} (\mathbf{H}_F^{-1} \Delta\mathbf{H}) \mathbf{H}_F^{-1} \quad (14)$$

and

$$\Delta\mathbf{p}^T \mathbf{H}_F^{-1} (\mathbf{H}_F^{-1} \Delta\mathbf{H})^{i-1} = \zeta_k (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} \mathbf{h}_k^T. \quad (15)$$

Therefore the expansion

$$(\mathbf{H}_F + \epsilon \Delta\mathbf{H})^{-1} = \sum_{i=0}^{+\infty} (-\epsilon)^i (\mathbf{H}_F^{-1} \Delta\mathbf{H})^i \mathbf{H}_F^{-1} \quad (16)$$

$$= \mathbf{H}_F^{-1} + \sum_{i=1}^{+\infty} (-\epsilon)^i (\mathbf{h}_k^T \boldsymbol{\delta}_k)^{i-1} (\mathbf{H}_F^{-1} \Delta\mathbf{H}) \mathbf{H}_F^{-1}. \quad (17)$$

From (8), (9) and (11), we have  $P_F = \mathbf{p}_F^T \mathbf{H}_F^{-1} \mathbf{c}$  and  $D_F = \frac{1}{\alpha A} \mathbf{d}^T \mathbf{H}_F^{-1} \mathbf{c} - 1$ . Therefore

$$\begin{aligned} & \frac{P_{F''} - P_F}{P_{F'} - P_F} \\ &= \frac{(\mathbf{p}_F + \epsilon \Delta\mathbf{p})^T (\mathbf{H}_F + \epsilon \Delta\mathbf{H})^{-1} \mathbf{c} - \mathbf{p}_F^T \mathbf{H}_F^{-1} \mathbf{c}}{(\mathbf{p}_F + \Delta\mathbf{p})^T (\mathbf{H}_F + \Delta\mathbf{H})^{-1} \mathbf{c} - \mathbf{p}_F^T \mathbf{H}_F^{-1} \mathbf{c}} \end{aligned} \quad (18)$$

$$\begin{aligned} & \frac{p_F^T [(H_F + \epsilon \Delta H)^{-1} - H_F^{-1}] c}{+\epsilon \Delta p^T (H_F + \epsilon \Delta H)^{-1} c} \\ &= \frac{p_F^T [(H_F + \Delta H)^{-1} - H_F^{-1}] c}{+\Delta p^T (H_F + \Delta H)^{-1} c} \end{aligned} \quad (19)$$

$$\begin{aligned} & \frac{p_F^T \left[ \sum_{i=1}^{+\infty} (-\epsilon)^i (h_k^T \delta_k)^{i-1} (H_F^{-1} \Delta H) H_F^{-1} \right] c}{-\Delta p^T \left[ \sum_{i=1}^{+\infty} (-\epsilon)^i (H_F^{-1} \Delta H)^{i-1} H_F^{-1} \right] c} \\ &= \frac{p_F^T \left[ \sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1} (H_F^{-1} \Delta H) H_F^{-1} \right] c}{-\Delta p^T \left[ \sum_{i=1}^{+\infty} (-1)^i (H_F^{-1} \Delta H)^{i-1} H_F^{-1} \right] c} \end{aligned} \quad (20)$$

$$\begin{aligned} & \frac{\sum_{i=1}^{+\infty} (-\epsilon)^i (h_k^T \delta_k)^{i-1} p_F^T (H_F^{-1} \Delta H) H_F^{-1} c}{-\sum_{i=1}^{+\infty} (-\epsilon)^i \zeta_k (h_k^T \delta_k)^{i-1} h_k^T c} \\ &= \frac{\sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1} p_F^T (H_F^{-1} \Delta H) H_F^{-1} c}{-\sum_{i=1}^{+\infty} (-1)^i \zeta_k (h_k^T \delta_k)^{i-1} h_k^T c} \end{aligned} \quad (21)$$

$$\begin{aligned} &= \frac{\sum_{i=1}^{+\infty} (-\epsilon)^i (h_k^T \delta_k)^{i-1}}{\sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1}} \\ &= \frac{\epsilon + \epsilon h_k^T \delta_k}{1 + \epsilon h_k^T \delta_k} \end{aligned} \quad (22)$$

$$\begin{aligned} &= \frac{\epsilon + \epsilon h_k^T \delta_k}{1 + \epsilon h_k^T \delta_k} \end{aligned} \quad (23)$$

and

$$\begin{aligned} & \frac{D_{F''} - D_F}{D_{F'} - D_F} \\ &= \frac{d^T (H_F + \epsilon \Delta H)^{-1} c - d^T H_F^{-1} c}{d^T (H_F + \Delta H)^{-1} c - d^T H_F^{-1} c} \end{aligned} \quad (24)$$

$$\begin{aligned} &= \frac{d^T \left( \sum_{i=1}^{+\infty} (-\epsilon)^i (h_k^T \delta_k)^{i-1} (H_F^{-1} \Delta H) H_F^{-1} \right) c}{d^T \left( \sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1} (H_F^{-1} \Delta H) H_F^{-1} \right) c} \end{aligned} \quad (25)$$

$$\begin{aligned} &= \frac{\sum_{i=1}^{+\infty} (-\epsilon)^i (h_k^T \delta_k)^{i-1}}{\sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1}} \end{aligned} \quad (26)$$

$$\begin{aligned} &= \frac{\epsilon + \epsilon h_k^T \delta_k}{1 + \epsilon h_k^T \delta_k}. \end{aligned} \quad (27)$$

Hence  $\frac{P_{F''} - P_F}{P_{F'} - P_F} = \frac{D_{F''} - D_F}{D_{F'} - D_F} = \frac{\epsilon + \epsilon h_k^T \delta_k}{1 + \epsilon h_k^T \delta_k} = \epsilon'$ , so that  $P_{F''} = (1 - \epsilon')P_F + \epsilon'P_{F'}$  and  $D_{F''} = (1 - \epsilon')D_F + \epsilon'D_{F'}$ . Moreover, it can be observed that  $\epsilon' = \frac{\epsilon + \epsilon h_k^T \delta_k}{1 + \epsilon h_k^T \delta_k}$  is a continuous nondecreasing function.

2) From the first part, we know  $\frac{P_{F''} - P_F}{P_{F'} - P_F} = \frac{D_{F''} - D_F}{D_{F'} - D_F} = \epsilon'$  and  $\epsilon'$  is a continuous nondecreasing function of  $\epsilon$ . When  $\epsilon = 0$ ,  $\epsilon' = 0$ . When  $\epsilon = 1$ ,  $\epsilon' = 1$ . Therefore when  $\epsilon$  changes from 0 to 1, the point  $(P_{F''}, D_{F''})$  moves on the line segment from  $(P_F, D_F)$  to  $(P_{F'}, D_{F'})$ . The slope of the line is

$$\begin{aligned} & \frac{D_{F'} - D_F}{P_{F'} - P_F} \\ &= \frac{\frac{1}{\alpha A} d^T (H_F + \Delta H)^{-1} c - \frac{1}{\alpha A} d^T H_F^{-1} c}{(p_F + \Delta p)^T (H_F + \Delta H)^{-1} c - p_F^T H_F^{-1} c} \end{aligned} \quad (28)$$

$$\begin{aligned} &= \frac{\frac{1}{\alpha A} d^T \left( \sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1} (H_F^{-1} \Delta H) H_F^{-1} \right) c}{\sum_{i=1}^{+\infty} (-1)^i (h_k^T \delta_k)^{i-1} p_F^T (H_F^{-1} \Delta H) H_F^{-1} c} \\ &\quad - \sum_{i=1}^{+\infty} (-1)^i \zeta_k (h_k^T \delta_k)^{i-1} h_k^T c \end{aligned} \quad (29)$$

$$\begin{aligned} &= \frac{\frac{1}{\alpha A} d^T H_F^{-1} \Delta H H_F^{-1} c}{p_F^T H_F^{-1} \Delta H H_F^{-1} c - \zeta_k h_k^T c} \end{aligned} \quad (30)$$

$$\begin{aligned} &= \frac{d^T H_F^{-1} \delta_k}{\alpha A (p_F^T H_F^{-1} \delta_k - \zeta_k)}. \end{aligned} \quad (31)$$

Theorem 1 indicates that the convex combination of scheduling strategies which are different only for one  $k$  will induce the convex combination of delay-power pairs. The slope of the line segment from  $(P_F, D_F)$  to  $(P_{F'}, D_{F'})$  is obtained in (31). Furthermore, we can have the following enhanced conclusions, whose strict proofs are omitted because of space limitation.

**Corollary 1.** The set of all feasible delay-power pairs is a polyhedron. The vertices of the polyhedron are all obtained by deterministic scheduling strategies.

Denote  $\mathcal{R}$  as the set of all possible delay-power pairs. Denote  $\mathcal{C}$  as the convex hull of delay-power pairs generated by deterministic scheduling strategies. Corollary 1 actually demonstrate that  $\mathcal{R} = \mathcal{C}$ . However, what we concern most is the optimal delay-power tradeoff curve  $\mathcal{L} = \{(P, D) \in \mathcal{R} | \forall (P', D') \in \mathcal{R}, \text{ either } P' \geq P \text{ or } D' \geq D\}$ . In other words,  $\mathcal{L}$  is the Pareto frontier of  $\mathcal{R}$ . Based on Corollary 1, immediately we can have that

**Corollary 2.** The optimal delay-power tradeoff curve is piecewise linear. The vertices of the curve are obtained by deterministic scheduling strategies.

Moreover, the scheduling strategies leading to the optimal delay-power tradeoff have the following property.

**Theorem 2.** The optimal scheduling strategy is threshold-based. That is to say, there exists  $(M + 1)$  thresholds  $k_0 \leq k_1 \leq \dots \leq k_M$ , one of which we denote as  $k_{m^*}$ , such that

$$\begin{cases} f_{k,m} = 1 & k_{m-1} < k \leq k_m, m \neq m^* \\ f_{k,m} = 1 & k_{m^*-1} < k < k_{m^*} \\ f_{k_{m^*}, m^*} + f_{k_{m^*}, m^*+1} = 1 \\ f_{k,m} = 0 & \text{else} \end{cases}. \quad (32)$$

For simplicity and unification, denote  $k_{-1} = -1$ . More specifically, we have  $k_0 = 0$ ,  $k_A = k_{A+1} = \dots = k_M = K$ .

The proof of Theorem 2 is also omitted because of space limitation. The theorem indicates that more data should be transmitted if the queue is longer. When  $k_{m-1} < t[n] < k_m$ ,  $m$  bits should be transmitted. Any optimal scheduling strategy  $F$  has at most two decimal elements  $f_{k_{m^*}, m^*}$  and  $f_{k_{m^*}, m^*+1}$ , while the other elements are either 0 or 1. The strategies leading to two adjacent vertices and their line segment are only different for  $t[n] = k_{m^*}$ . For the vertex with larger delay and less power,  $f_{k_{m^*}, m^*} = 1$ . For the vertex with smaller delay and more power,  $f_{k_{m^*}, m^*+1} = 1$ . This confirms that the vertices are obtained by deterministic strategies.

Based on Corollary 2 and Theorem 2, we know that the optimal delay-power tradeoff curve is obtained by threshold-based strategy. The vertices of the piecewise linear curve are obtained by deterministic strategy. Therefore Algorithm 1 is proposed to effectively obtain the optimal delay-power tradeoff

---

**Algorithm 1** Obtain the Optimal Delay-Power Tradeoff

---

**initialize:**

- 1) Set  $\mathbf{F}$  as a threshold-based deterministic strategy where  $k_m = \min\{m, A\}$  for  $0 \leq m \leq M$
- 2) Calculate the corresponding average delay  $d_c$  and average power consumption  $p_c$
- 3) Denote set  $\mathcal{F}_c = \{\mathbf{F}\}$

**while**  $\mathcal{F}_c$  is not null **do**

- 1)  $\mathcal{F}_p = \mathcal{F}_c$ ,  $d_p = d_c$ ,  $p_p = p_c$ ,  $\mathcal{F}_c = \emptyset$ ,  $s = +\infty$
- 2) For every strategy in  $\mathcal{F}_p$ , we take turns to increase each one of its threshold, thus generate a new strategy  $\mathbf{F}$  each time. For every legal scheduling strategy  $\mathbf{F}$ :  
**if** its corresponding average delay  $d$  and average power  $p$  satisfies  $d \geq d_p$  and  $p < p_p$  and  $\frac{d-d_p}{p_p-p} < s$   
**then**  $\mathcal{F}_c = \{\mathbf{F}\}$ ,  $s = \frac{d-d_p}{p_p-p}$ ,  $d_c = d$ ,  $p_c = p$   
**else if** its corresponding average delay  $d$  and average power  $p$  satisfies  $d \geq d_p$  and  $p < p_p$  and  $\frac{d-d_p}{p_p-p} = s$   
**then** add  $\mathbf{F}$  into set  $\mathcal{F}_c$   
**end if**
- 3) Draw the line segment connecting  $(p_p, d_p)$  and  $(p_c, d_c)$

**end while**

---

curve. In the algorithm, we start from the bottom right vertex of the curve, and search for the next vertex on the optimal tradeoff curve one by one.

## V. LINEAR PROGRAMMING FORMULATION

In the above section, we analyse the optimal delay-power tradeoff as the Pareto frontier of the feasible delay-power region. In this section, we investigate the tradeoff curve from an optimization perspective.

Based on (8) (9) and (11), to achieve the minimum delay given a certain power constraint  $P_{th}$ , we formulate the following optimization problem

$$\min_{\mathbf{F}, \boldsymbol{\pi}} \quad \frac{1}{\alpha A} \mathbf{d}^T \boldsymbol{\pi} - 1 \quad (33)$$

$$\text{s.t.} \quad \mathbf{p}_F^T \boldsymbol{\pi} \leq P_{th} \quad (34)$$

$$\mathbf{H}_F \boldsymbol{\pi} = \mathbf{c} \quad (35)$$

$$\boldsymbol{\pi} \succeq \mathbf{0} \quad (36)$$

$$f_{k,m} = 0 \quad \forall k-m < 0 \text{ or } k-m > K-A \quad (37)$$

where  $\boldsymbol{\pi} \succeq \mathbf{0}$  means  $\boldsymbol{\pi}$  is componentwise nonnegative. By varying the value of  $P_{th}$ , the corresponding optimal delay will be obtained. Hence the optimal delay-power tradeoff curve can be acquired.

Define  $x_{k,m} = \pi_k f_{k,m}$ . From (3), (9) and (11), we can express the average delay and power using  $x_{k,m}$  as

$$P = \sum_{k=0}^K \sum_{m=0}^M P_m x_{k,m} \quad (38)$$

and

$$D = \frac{1}{\alpha A} \sum_{k=0}^K k \sum_{m=0}^M x_{k,m} - 1. \quad (39)$$

From global equilibrium equations

$$\begin{aligned} & \sum_{l=\max\{0, k-A\}}^{k-1} \sum_{m=0}^{l+A-k} \alpha \pi_l f_{l,m} \\ &= \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+1}^{r-k+A} (1-\alpha) \pi_r f_{r,m} \\ &+ \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+A+1}^M \pi_r f_{r,m} \quad k = 1, \dots, K \end{aligned} \quad (40)$$

we have

$$\begin{aligned} & \sum_{l=\max\{0, k-A\}}^{k-1} \sum_{m=0}^{l+A-k} \alpha x_{l,m} = \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+A+1}^M x_{r,m} \\ &+ \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+1}^{r-k+A} (1-\alpha) x_{r,m} \quad k = 1, \dots, K. \end{aligned} \quad (41)$$

The hazard of overflow or underflow can be eliminated by setting  $f_{k,m} = 0$  if  $k-m < 0$  or  $k-m > K-A$ , hence  $x_{k,m} = 0$  if  $k-m < 0$  or  $k-m > K-A$ . Moreover, since  $\sum_{k=0}^K \pi_k = 1$ , we have  $\sum_{k=0}^K \sum_{m=0}^M x_{k,m} = 1$ . Therefore the original optimization problem can be transformed to

$$\min \quad \frac{1}{\alpha A} \sum_{k=0}^K k \sum_{m=0}^M x_{k,m} - 1 \quad (42)$$

$$\text{s.t.} \quad \sum_{k=0}^K \sum_{m=0}^M P_m x_{k,m} \leq P_{th} \quad (43)$$

$$\begin{aligned} & \sum_{l=\max\{0, k-A\}}^{k-1} \sum_{m=0}^{l+A-k} \alpha x_{l,m} \\ &= \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+A+1}^M x_{r,m} \\ &+ \sum_{r=k}^{\min\{k+M-1, K\}} \sum_{m=r-k+1}^{r-k+A} (1-\alpha) x_{r,m} \quad k = 1, \dots, K \end{aligned} \quad (44)$$

$$\sum_{k=0}^K \sum_{m=0}^M x_{k,m} = 1 \quad (45)$$

$$x_{k,m} = 0 \quad \forall k-m < 0 \text{ or } k-m > K-A \quad (46)$$

$$x_{k,m} \geq 0 \quad \forall 0 \leq k-m \leq K-A. \quad (47)$$

It can be observed that this is a linear programming problem. Based on mature algorithms such as simplex method or interior point method, the optimization can be well solved.

## VI. NUMERICAL RESULTS

In this section, we validate our theoretical conclusions by conducting numerical computation and simulation. It will be confirmed that the optimal delay-power tradeoff curve obtained by the Pareto frontier of the feasible delay-power

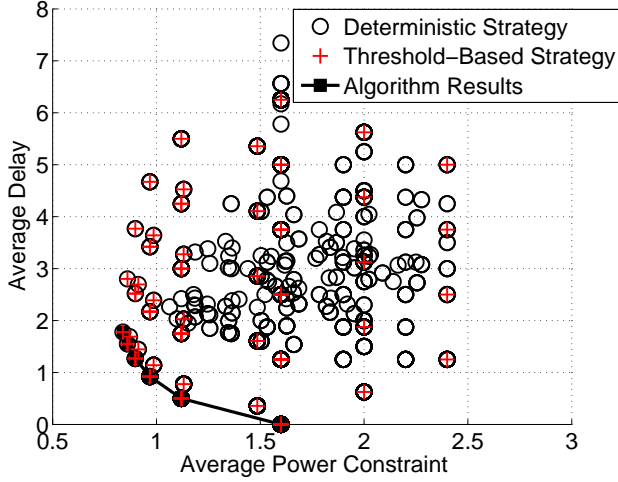


Fig. 3. Delay-Power Pairs Generated by Deterministic Strategies

region and the one obtained by linear programming are the same.

In Fig.3, we show all the delay-power pairs generated by deterministic strategies. The parameters are  $K = 7$ ,  $A = 2$ ,  $M = 3$ ,  $\alpha = 0.4$ ,  $P_0 = 0$ ,  $P_1 = 1$ ,  $P_2 = 4$ ,  $P_3 = 9$ . Hence the feasible delay-power region is the convex hull of all the points. Within these delay-power pairs, we mark the ones that are corresponding to threshold-based strategies with '+' markers. As we can see, Algorithm 1 obtains the Pareto frontier of the feasible delay-power region, which is the optimal delay-power tradeoff curve. The optimal tradeoff curve is piecewise linear, whose vertices are all obtained by threshold-based deterministic strategies.

In Fig.4 where  $K = 7$ ,  $A = 2$ ,  $M = 3$ ,  $P_0 = 0$ ,  $P_1 = 1$ ,  $P_2 = 4$ ,  $P_3 = 9$ , we demonstrate that the optimal delay-power tradeoff curve obtained by linear programming completely overlap the optimal tradeoff curve generated by Algorithm 1. The results are further validated by Monte-Carlo simulation. As is expected, the average delay decreases when the average power consumption increases. When the power constraint is no less than  $\alpha P_A$ , which corresponds to the initial strategy of Algorithm 1, the average delay decreases to 0 because any data will be immediately transmitted. With  $\alpha$  increasing, the curve gets higher because of the heavier workload.

## VII. CONCLUSION

In this paper, we investigate the buffer-aware probabilistic scheduling strategies and the corresponding average delay and power consumption. Specifically, random data arrival and adaptive transmission rate are considered. The average delay and power consumption are analysed by Markov reward process. Based on this, we discover that the feasible delay-power region is a polyhedron. The optimal delay-power tradeoff curve, as the Pareto frontier of the polyhedron, is piecewise linear, whose vertices are obtained by deterministic strategies. Furthermore, the optimal delay-power tradeoff is achieved

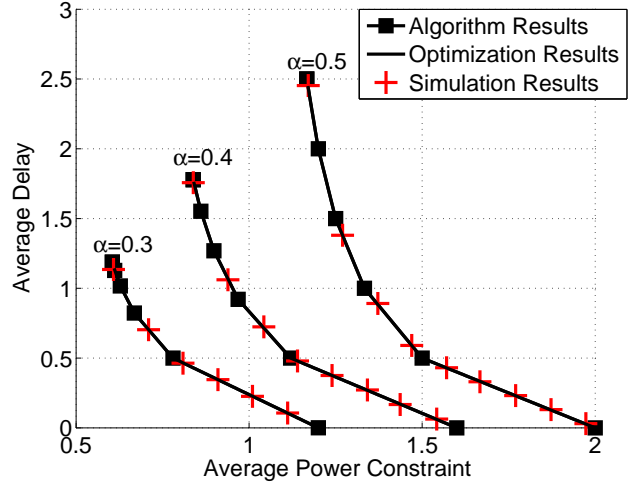


Fig. 4. Optimal Delay-Power Tradeoff Curves

by threshold-based strategies, thus an effective algorithm is proposed to obtain the optimal tradeoff curve. Moreover, seen from another angle, we formulate a linear programming problem to minimize the average delay given a fixed average power constraint. The optimal delay-power tradeoff curves obtained by the algorithm and the optimization problem match perfectly well, which are further validated by Monte-Carlo simulation.

## REFERENCES

- [1] B. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," in *Proc. 37th Allerton Conf. Commun. Control, Comput., Monticello, IL*, 1999, pp. 709–717.
- [2] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.
- [3] M. Goyal, A. Kumar, and V. Sharma, "Power constrained and delay optimal policies for scheduling transmission over a fading channel," in *Proc. IEEE INFOCOM*, 2003, pp. 311–320.
- [4] I. Bettesh and S. Shamai, "Optimal power and rate control for minimal average delay: The single-user case," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4115–4141, 2006.
- [5] B. Ata, "Dynamic power control in a wireless static channel subject to a quality-of-service constraint," *Oper. Res.*, vol. 53, no. 5, pp. 842–851, 2005.
- [6] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 487–499, 2002.
- [7] M. A. Zafer and E. Modiano, "A calculus approach to energy-efficient data transmission with quality-of-service constraints," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 898–911, 2009.
- [8] J. Lee and N. Jindal, "Energy-efficient scheduling of delay constrained traffic over fading channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1866–1875, 2009.
- [9] R. Negi and J. M. Cioffi, "Delay-constrained capacity with causal feedback," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2478–2494, 2002.
- [10] W. Chen, Z. Cao, and K. B. Letaief, "Optimal delay-power tradeoff in wireless transmission with fixed modulation," in *Proc. IEEE IWCLD*, 2007, pp. 60–64.
- [11] X. Chen and W. Chen, "A joint channel-aware and buffer-aware scheduling for energy-efficient transmission over fading channels with long coherent time," in *Proc. IEEE GlobalSIP*, 2014, pp. 103–107.
- [12] —, "Joint channel-buffer aware energy-efficient scheduling over fading channels with short coherent time," in *Proc. IEEE ICC*, 2015, to appear.